

## **HYBRID TESTER ARCHITECTURE**

### **FIELD OF THE INVENTION**

The invention relates generally to automatic test equipment, and more particularly to a hybrid semiconductor tester architecture for testing semiconductor devices.

### **BACKGROUND OF THE INVENTION**

Automatic test equipment provides the ability for semiconductor device manufacturers to test each and every device made. By testing each device, the manufacturer can sort devices having like speeds, and/or separate failed devices from passing devices. In this manner, the manufacturer is able to confidently put fully functioning devices into the marketplace.

Testing a semiconductor device typically involves applying signals to specified pins according to precise timings, and detecting the device outputs in response to the applied signals. The detected output signals are then compared to expected values to determine whether the device operated within a specified range of parameters. Many of the high-level design considerations for a semiconductor tester focus on the manner of applying and detecting signals to and from the device-under-test (DUT).

With this in mind, semiconductor tester architectures usually fall within one of two types, shared resource, or tester-per-pin. In a conventional tester-per-pin architecture, such as that shown in Figure 1, each tester channel includes separate tester resources for assignment to one pin of a DUT. Those resources include the necessary timing circuitry 12, pattern generation circuitry 14, formatting circuitry 16, and pin electronics 18 for applying signals to, or receiving signals from a pin of the DUT 20. A failure processing circuit handles the evaluation of expected test data versus actual data. This architecture is highly desirable when testing complex logic devices that need flexibility in testing the individual device pins.

Shared resource architectures, on the other hand, are most often found in memory testers, where multiple devices are tested simultaneously to maximize throughput and minimize test costs. A conventional shared resource architecture is shown in Figure 2, illustrating shared resources for a 2048 channel tester. In this example, sixty-four memory devices 40 (each having thirty-two pins) may be tested with, for example, thirty-two copies 40 of the timing 42, formatting 44, and pattern

generation (or data generation) circuitry 46. Fanout circuitry 48 distributes the shared formatted tester signals in parallel to the multiple DUTs 30.

After testing the devices on a shared resource tester, the failure data is typically stored in a bit-image failure memory known as a catchram 50, and  
5 subsequently downloaded to redundancy analysis circuitry 52. After the data is analyzed by the redundancy analyzers, a repair solution is sent to a laser repair station 60, along with the failed devices, where the redundant lines identified for use in repairing the device are programmed into operation.

While the conventional shared resource architecture works well for many test  
10 applications, and are often far less costly than their per-pin counterparts, different DUTs typically have bits that fail in different rows and/or columns. Consequently, the repair schemes for different devices will be different.

Moreover, some device manufacturers choose to repair devices on the tester without delays inherent in transmitting failure data to separate repair stations.  
15 Repairing DUTs on the tester generally involves writing unique repair data after testing to each device to activate internal device fuses corresponding to the desired repair solution. However, without having the ability to write the individual data for each device pin to each device, the device manufacturer cannot repair the devices in this manner. As a result, many manufacturers opt for the more expensive per-pin  
20 tester architectures in order to accomplish on-tester device repair.

What is needed and currently unavailable is a low-cost hybrid tester architecture solution that provides the per-pin benefits of unique device failure identification with the shared-resource advantages of low-cost. The hybrid tester architecture of the present invention satisfies these needs.

### **SUMMARY OF THE INVENTION**

The hybrid tester architecture of the present invention provides a way to preserve the low cost attributes of shared resource testers without sacrificing data generation flexibility. To realize the foregoing advantages, the invention in one form  
5 comprises a hybrid tester architecture for testing a plurality of semiconductor devices in parallel. The hybrid tester architecture includes per-pin formatting circuitry having data input circuitry and clock input circuitry and shared timing circuitry coupled to the clock input circuitry. The shared timing circuitry generates programmed timing signals. Per-pin data circuitry couples to the data input circuitry and generates drive  
10 data and expected data values associated with each individual device pin. The per-pin formatting circuitry responds to the programmed timing signals to produce tester waveforms in accordance with the per-pin data.

Other features and advantages of the present invention will be apparent from the following detailed description when read in conjunction with the accompanying  
15 drawings.

**BRIEF DESCRIPTION OF THE DRAWINGS**

The invention will be better understood by reference to the following more detailed description and accompanying drawings in which

5      FIG. 1 is a high-level block diagram of a conventional per-pin tester architecture;

FIG. 2 is a high-level block diagram of a conventional shared resource tester architecture; and

10      FIG. 3 is a high-level block diagram of a hybrid tester architecture according to one form of the present invention.

## DETAILED DESCRIPTION OF THE INVENTION

The hybrid tester architecture of the present invention provides a low-cost way to test semiconductor devices in parallel while preserving device data uniqueness. This is done by employing shared resources for certain tester functions, and per-pin  
5 resources for other functions. As a result, on-tester device repair is achievable with a low-cost tester.

Referring now to Figure 3, the hybrid tester architecture according to one form of the present invention, generally designated 100, includes a computer workstation 102 and a tester body 104 (in phantom). The tester body includes channel resources  
10 that define a hybrid tester architecture sufficient to test a plurality of DUTs 106 in parallel. A failure memory 115 and redundancy analysis circuitry 117 cooperate with the channel resources to test and repair the devices as more fully described below.

Further referring to Figure 3, the hybrid channel resources include shared-pin timing circuitry 108, per-pin formatting circuitry 110, and per-pin data generation  
15 circuitry 112. The shared-pin timing circuitry comprises a set of thirty-two modules, or "copies" of the tester timing system resources for generating signals in parallel through the per-pin formatting circuitry 110 to the same "like" pin for all of the DUTs 106.

The per-pin formatting circuitry 110 preferably comprises digital circuitry in  
20 the form of a plurality of flip-flops (not shown), each having a data input (not shown) and a clock input (not shown) for clocking in data in response to the timing signals. In this manner, the timing signal may be synchronized to the data, as is well known in the art.

With continued reference to Figure 3, the per-pin data generation circuitry 112  
25 preferably comprises a plurality of dedicated memory blocks sufficient to store unique data for each pin of the DUTs 106. In the example where sixty-four devices are tested in parallel, each device having thirty-two pins, 2048 memory blocks are available to provide the per-pin data resources. Preferably, the memory blocks are realized as RAM memories to minimize costs.

Consistent with the example above, each of the per-pin memory blocks feeds  
30 its output to a corresponding per-pin formatting circuit 110 (for a total of 2048 blocks). As noted above, the per-pin data gets clocked through the per-pin formatting circuitry by the shared-pin timing circuitry 108.

Prior to operation, the per-pin data generation resources 112 are programmed  
35 to provide identical data values for each of the thirty-two sets of shared timing

resources 108. In this sense, the data generation circuitry may be viewed as "shared" at this stage of testing. This is because each of the DUTs 106 are identical, and go through the same initial test patterns in order to determine faults.

5 As testing progresses, each of the DUTs 106 undergoes write and read operations so that the tester can determine if any of the DUTs have faulty cells, and if so, where those cells lie. The failure data for each DUT 106 is routed through respective databuses 111 (shown in phantom) and stored in the failure memory 115, as is well known in the art.

10 Following completion of the test, the failure data from the failure memory 115 is fed to the on-tester redundancy analysis circuitry 117. The circuitry analyzes the faulty data for each DUT and assigns redundant rows and columns (built into each DUT) to "fix" the device. Each redundancy solution is unique for a particular DUT (the redundant row/column addresses) and generally involves programming the DUT to incorporate certain row and column address lines. For on-tester device repair, the  
15 solution data is fed back to the computer workstation 102. Once the solutions are known, corresponding test patterns are created and fed to the per-pin data resources 112 to then generate write addresses for programming activation of the desired redundancy elements for each device.

20 If desired, a second test may be performed on the devices to verify the acceptability of the repair solutions for each device.

Those skilled in the art will appreciate the many benefits and advantages afforded by the present invention. Of significant importance is the preservation of low cost for testing semiconductor devices in parallel through use of shared timing and formatting resources, yet maximizing tester flexibility through the  
25 implementation of per-pin data resources.

While the invention has been particularly shown and described with reference to the preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention. For example, while the description herein has  
30 focused on memory testers and memory devices, the invention may be utilized on logic devices where per-pin timing is unnecessary. This provides a very low-cost solution to testing logic devices.